Project Report: HelpMate AI

By - Raja Kalavala

HelpMate AI – RAG Insurance Assistant
Simplifying Insurance Document Queries using Retrieval-Augmented Generation and OpenAI's GPT Models

---

1. Introduction

The HelpMate AI project is an intelligent solution designed to streamline the process of understanding insurance policies. Insurance documents are often long, complex, and difficult for users to navigate. HelpMate addresses this challenge using Retrieval-Augmented Generation (RAG), integrating natural language generation capabilities from advanced AI models like GPT-4 with efficient document search and retrieval.

---

2. Problem Statement

Users frequently struggle to extract key information from insurance documents due to their technical language and structure. This leads to delays, frustration, and inefficient customer service. The project aims to resolve this by enabling users to query insurance documents using natural language and receive accurate, clear responses instantly.

---

3. Project Objectives

- Simplify access to insurance information.
- Enable natural language queries over document content.
- Leverage RAG to combine document retrieval with LLM-powered summarization.
- Deliver real-time, accurate, and context-rich answers.

---

4. Key Features

- **Accurate Information Retrieval**: Uses embeddings to find the most relevant parts of documents.
- **Natural Language Understanding**: Converts user queries into concise, helpful answers.
- **AI-Powered Insights**: Utilizes GPT-based models for better explanation and contextualization.
- **Fast Performance**: Integrated ChromaDB with caching for speed and efficiency.
- **Cross-Encoder Re-ranking**: Ranks results for improved relevance.
- **Custom System Prompting**: Uses few-shot learning and instructions to fine-tune responses.
- **User-Friendly Interface**: Built with usability in mind for non-technical users.

---

5. System Design

The HelpMate AI system is constructed using a modular Retrieval-Augmented Generation (RAG) pipeline, optimized for scalability, clarity, and high-performance natural language querying over insurance documents.

High Level Architecture:

---

6. Technologies Used

Component	Technology
Programming Language	Python 3.8+
Notebook Environment	Jupyter Notebook
Libraries/Frameworks	Transformers, ChromaDB, PDFplumber
AI Models	GPT-4, GPT-4o, Gemini API (optional)
Vector Store	ChromaDB (with optional caching)
Deployment Option	Local / Cloud (Docker supported)

---

7. Installation & Setup

Prerequisites

- Python 3.8 or higher
- Docker (optional)

Steps

1. Clone the repository:
   ```
   git clone https://github.com/RajaKalavala/HelpMateAI.git
   ```
2. Navigate to the project folder.
3. Install dependencies:
   ```
   pip install -r requirements.txt
   ```
4. Obtain an OpenAI API key and configure it in the code.
5. Launch the notebook:
   Open `helpmateAI.ipynb` in Jupyter.

---

8. Example Use Cases

- *“What is covered under my health insurance policy?”*
- *“How can I file a claim for vehicle insurance?”*

These queries are answered using actual policy documents through RAG pipelines.

---

9. Challenges Faced & Solutions

Issue	Solution

PDF parsing tools like PyPDF2/PDFMiner were insufficient	Switched to PDFplumber for accurate text & table extraction
Extracting data from tables	Redesigned logic to use PDFplumber's table handling
Repeated embedding slowing down ChromaDB	Added cache layer to avoid re-embedding
Irrelevant search results	Integrated a cross-encoder reranker for better passage relevance
Poor answer quality	Redesigned system prompts with few-shot examples and structured instructions

---

10. Future Enhancements

- Add support for more LLMs (Claude, Gemini, Hugging Face models)
- Add multiple vector store options (Weaviate, Pinecone)
- Extend feature set with multi-document QA, analytics, and voice-based interface

---

11. Documentation

This project is built on top of well-documented tools. Please refer to:

- ChromaDB: https://docs.trychroma.com/
- PDFplumber: https://pypi.org/project/pdfplumber/0.1.2/
- Sentence Transformers: https://www.sbert.net/docs/
- OpenAI: https://platform.openai.com/docs/